Towards a coherent enterprise modelling landscape

Marija Bjeković[1,2,3], Hend*erik* A. Proper[1,2,3], and Jean-Sébastien Sottet[1,3]

[1] Public Research Centre Henri Tudor, Luxembourg, Luxembourg
[2] Radboud University Nijmegen, Nijmegen, the Netherlands
[3] EE-Team**, Luxembourg, Luxembourg
{marija.bjekovic,erik.proper,jean-sebastien.sottet}@tudor.lu

Abstract. When modelling enterprises, for instance as part of an enterprise (re)engineering effort, one typically uses a range of models. These models differ in their intended purpose in terms of the domain which the model should pertain to and the intended usage of the model by its audience. The models are therefore generally created in purpose-specific modelling languages; i.e. not just *domain-specific* languages.
While using purpose-specific modelling languages has clear benefits in terms of the suitability of the language to a purpose at hand, there is also a downside to it. As each of the resulting enterprise models refers to (different aspects of) the same (version of the) enterprise, it is desirable to maintain coherence across the different models. The use of a wide range of purpose-specific models (and modelling languages) can easily lead to a fragmentation of the modelling landscape; i.e. a break up of coherence. This leads to a natural polarity between coherence and purpose-specificity. We argue that this polarity requires careful management, but first and foremost a better understanding.
To cope with, or avoid, the consequences of fragmentation, different strategies to achieve the integration of models and languages used in enterprise modelling have been suggested in the literature. These approaches, however, focus mainly on syntactic aspects of the models, while sometimes indeed including their (formal) semantics, and only to some extent their pragmatics.
The aim of the research, reported on in this paper, is to achieve a deeper understanding of the needs and challenges of the use of different models in enterprise modelling.

Key words: enterprise modelling, model integration, modelling languages

1 Introduction

When deliberately changing (parts of an) enterprise, one generally uses models to understand the current situation of the enterprise, analyse the problems/challenges with regard to the current situation, sketch potential future scenario's, and design selected future desired states of the enterprise, etc. We use the term *enterprise modelling landscape*, or simply *modelling landscape*, to refer to the variety of models (and corresponding purpose-specific modelling languages) used in such efforts. The developing

** The Enterprise Engineering Team (EE-Team) is a collaboration between Public Research Centre Henri Tudor, Radboud University Nijmegen and HAN University of Applied Sciences (www.ee-team.eu).

field of enterprise engineering [1, 2] also strongly promotes the use of a model-enabled approach to the transformation of enterprises.

When modelling enterprises in an enterprise engineering context, one must do so from the perspective of different domains, such as business processes, value exchanges, products and services, information systems, etc. The fact that enterprise modelling needs to deal with different domains, within the context of a specific enterprise is certainly not new. In the field of information systems engineering, the use of a multi-perspective approach has long since been advocated, e.g. [3, 4]. In the case of enterprise modelling, however, the number of domains to be included increases. For example, enterprise architecture frameworks suggest a much wider range of views that look beyond the traditional business-to-IT stack [5, 6, 7].

We argue that the plethora of models used in enterprise engineering efforts is brought about by the fact that models are needed for different *purposes*. In our current understanding, and based on our earlier work in e.g. [8, 9], we see the purpose of a model as a combination of:

1. the domain which the model should pertain to (e.g. different aspects and/or versions of the enterprise, the scope, granularity, etc.) and
2. the planned usage of the model (e.g. analysis, sketching, contracting, execution, etc.) by its intended audience.

In other words, the *purpose* of a specific model is to capture some *domain* to enable some *usage* by its audience.

Ideally, such models are created in a *purpose-specific* modelling language that tunes the modelling constructs of the language to the domain to be modelled, as well as adjust the precision/form of the medium, syntax and semantics of the language to the intended usage of the models. In practical modelling situations we have observed how, depending on the modelling purpose at hand, generic modelling language, such as UML [10] or ArchiMate [11], are used in different ways with regard to the 'discipline' with which the syntax and semantics of the generic language is obeyed to. In our view, this essentially leads to purpose-specific 'variations' of the same original generic modelling language (differing in their syntactic and semantic restrictions).

The notion of purpose-specific modelling language is certainly related to the notion of domain-specific languages [5, 12]. We argue, however, that the intended usage of the model also has a key role to play in tuning modelling languages to the needs at hand. We also acknowledge the fact that the notion of *model purpose* is related to the notion of *model quality* [13, 9]. We will revisit this relationship in the remainder of the paper. It is also important to realize that *model purpose* is different from *modelling purpose* [13, 9]. The purpose of a model is indeed dependent on the modelling purpose, but we see them as being different phenomena. In this paper, we limit our discussions to *model* purpose.

Since all of the models of an enterprise modelling landscape provide different views on the *same* enterprise, it is quite natural (and in line with an engineering perspective) to expect that the sets of models form a coherent whole; i.e. linked where relevant and consistent as a whole. Having such coherence among models also enables cross-cutting and impact of change analysis, traceability analysis, etc. So, while using purpose-specific modelling languages has clear benefits in terms of suitability of the language (and models) to a purpose at hand, there is also a potential downside to it. A plethora of purpose-

specific models can easily lead to a fragmentation of the modelling landscape; i.e. a break up of coherence, also addressed as a "*Tower of Babel situation*" [14]. This leads to a natural polarity between coherence and purpose-specificity. A number of strategies has been suggested to achieve integrated use of models and languages used in enterprise modelling, e.g. [5, 15]. We argue that this polarity deserves careful management, but first and foremost requires more research to better understand the forces at work.

In this paper, which is the result of an ongoing research, we present our current understanding of fundamental challenges related to coherent enterprise modelling landscapes. We will start in Section 2 by exploring the polarity between coherence and purpose-specificity in more detail, with the aim of gaining a better appreciation of the forces that are involved. We will then continue in Section 3 by exploring some of the existing strategies to manage this polarity. This discussion is then used as a base to develop the first-most elements of a theory on managing the coherence of modelling landscapes in Section 4.

2 Fragmentation of enterprise modelling landscapes

As already discussed, enterprise modelling is likely to involve a plethora of purpose-specific models covering different aspects of the enterprise. At the same time, the diversity of models/languages increases the risk of fragmenting the modelling landscape. This section explores the forces having a potential fragmenting effect on the modelling landscape, by taking two main ingredients of a model's purpose as a starting point. We discuss our current understanding of the "fragmenting forces", as exerted by each of these ingredients respectively.

2.1 Domain-specificity based fragmentation

The perspective from which the enterprise is to be modelled is a major force of potential fragmentation. When modelling enterprises, several dimensions exist in which to identify distinct domains (aspects, viewpoints, perspectives) to model. This potentially leads to even so many domain-specific models and languages. To illustrate the diversity, without having the ambition yet to provide an orthogonal set of dimensions, consider the following possible dimensions:

Intervention design – This concerns the motivation/rationalisation of a desired intervention[3] (e.g. transformation or development effort) in an enterprise. Examples of different models along this dimension include:

- Models capturing the goals/motivations for changing the existing enterprise.
- Models capturing the requirements on the desired (direction of) change of the enterprise.

[3] Note that since an enterprise is a socio-technical system that, by its nature, has a tendency to change due to the initiatives of the humans that ultimately make up the enterprise, we prefer to use the term *intervention* rather than *system development* or even *implementation*. The term intervention more clearly signifies the fact that an enterprise is not just a technological artefact.

– Models (plans) of an intervention in the current enterprise to change it in the desired direction.

Within the plans for the intervention, one may also distinguish between different time horizons. In other words, what one might want to achieve in the next year, in five years from now, and beyond.

Enterprise design – This dimension deals with the motivation/rationalisation of the (existing/planned) design of an enterprise. In this dimension, one could for example distinguish between:
– Models capturing the goals/motivation for owning/using the (parts of the) enterprise.
– Models capturing the requirements on the enterprise that follow from this.
– Models capturing the design of the enterprise (meeting the requirements).

Design domains – This dimension concerns the domains that are considered relevant to the design of an enterprise, e.g. Zachman framework cells [4], the distinction between different levels of implementation specificity in Capgemini's Integrated Architecture Framework [16], the distinction between business, application and technology layer in ArchiMate [17] and TOGAF [18], etc. This dimension also includes 'cross cutting' domains such as security and governance as in e.g.[16], and different 'projections' on design aspects relevant to specific concerns/stakeholders using viewpoints [11].

Granularity – This concerns different granularities at which one might want to model parts of the enterprise. Depending on the specific aspect of the enterprise, different ways to identify levels or granularity might be relevant. For example, in the case of process modelling, one could distinguish:
– Level of key processes, without any triggering relationships between them.
– Level of major sub-processes, with some overall triggering relationships.
– Level of specific work tasks, with complete triggering relationships, including splits and joins.

Governance domains – This concerns the domains from which an enterprise wants to govern itself. As argued in the GEA [7] , these are not the same as the design domains. The design domains are typically formulated from an "engineering" perspective (blueprint thinking), while the governance domains are formulated more from an organizational and political perspective, e.g. human resourcing, compliance, acquisition, marketing, etc. These dimensions are also highly organization specific.

2.2 Usage-specificity based fragmentation

Models are created with an intended usage in mind, e.g. analysis, sketching, contracting, forecasting, simulation, execution, etc. The intended usage of the model by some audience will have a direct impact on the requirements on the modelling language used to capture the model [19, 8]. This invites more variety in models/languages used, adding to the potential fragmentation of modelling landscape. We suggest to identify:

Restriction of notation – refers to the level of restriction that is put on the notation that can be used to represent the model on a medium. The medium itself can for example be restricted to a specific form, such as graphical, textual, or video, but the notation in general can also be restricted in terms of fonts, icons and layout rules. See [20] for the role of notation in modelling.

Restriction of syntax – concerns the level of syntactic restrictions that may be put forward by the modelling language used. For example, one might consider "free format" drawings or text on one hand, and UML diagrams or text-based specification languages on the other extreme.

Restriction of semantics – refers to the extent to which a language is to be used with (an enforced!) formalized semantics. Formality in this context refers to the fact that the modelling language (graphical or textual) has an underlying semantics in some mathematical domain. See e.g. [21] for an elaborate discussion of formalizing the semantics of modelling language, and its relationship to the purpose of the models.

3 Managing coherence in enterprise modelling

Different strategies can be used to manage this potential fragmentation of the modelling landscape. In this section we discuss some of the strategies suggested in the literature, as well as their involved trade-offs.

3.1 Unified language

A classical approach to enable integrated modelling of the enterprise would involve relying on an all-encompassing and *unified* modelling language, to integrate all the relevant perspectives on the enterprise. This approach essentially boils down to preventing the fragmentation from occurring in the first place: it involves a single and stable language with an a priori defined set of concepts and their links, to be used uniformly. This would also lead to the standardisation of the language, i.e. vocabulary used for modelling, which in turn should facilitate knowledge transfer and communication about the system being engineered.

Such a line of reasoning can for instance be observed in the definitions of UML [10] for software design modelling, or ArchiMate [17] for enterprise architecture modelling. However, in the context of enterprise modelling, the feasibility of a unified language approach is questionable. First of all, it is nearly impossible to a priori identify which domains (and modelling concepts) should be part of an integrated language for enterprise modelling. Furthermore, the relevance of different domains is also highly situation-dependent. For example, different perspectives may be relevant for different enterprises, or even in different transformation projects of the same enterprise, or new perspectives may become relevant as the result of the evolution of the enterprise. A language such as ArchiMate was designed [11] to deal with this by enabling users to define their own viewpoints; i.e. essentially purpose-specific modelling languages where the model (the "view") is derived from the unified/integrated model. At the same time, however, one can see how there is a drive for the ArchiMate language as a whole to be extended with additional domains, the move from the ArchiMate 1.0 standard to the ArchiMate 2.0 standard [17] included two additional domains (motivation and migration). Further integration between TOGAF and ArchiMate is likely to lead to even more extensions, while extensions dealing with e.g. value modelling, are also considered. *Where will it stop?*

Secondly, in dealing with an enterprise, one has to acknowledge the heterogeneity of communities and their (sub)languages [22, 23]. In such a context, imposing the single unified language for modelling the enterprise is likely to cause the conceptual misunderstandings around the resulting models, and a lot of time and effort would have to be put in resolving them.

3.2 Federated languages

A more flexible strategy results when allowing the co-existence of different modelling languages to model different perspectives on an enterprise. Essentially, this approach acknowledges the benefits of focused modelling languages for particular purposes. The integration strategy here consists in seeking to establish the links between these different languages and models used, in general having the ambition of interoperability of modelling languages at syntactic and semantic levels. For example, the MEMO [5] framework provides a common meta-language (i.e. the MEMO meta-metamodel) for all the special purpose modelling languages. Any special purpose language can be included in the framework, once it is expressed in the common meta-language. The integration of languages is achieved by their sharing of common conceptual foundation.

The Unified Enterprise Modelling Language (UEML) [15] has the ambition of making languages definitions semantically interoperable, and in that way facilitate the integrated use of models in enterprise modelling. While it also allows the inclusion of new modelling languages in the framework, this approach requires full formal precision of all the enterprise modelling languages, regardless of the purpose they are intended for. The core of the UEML approach lies in a common and evolving ontology [24], in which the modelling constructs of the modelling languages are to be precisely described This approach focuses on precisely describing (only) the *type semantics* [25] of the language constructs based on their specifications. However, as already discussed, one can observe the existence of different purpose-specific syntactic and semantic variations of the same original generic modelling language. It is our belief that the pragmatics of languages, and not only their specifications, should be considered within the efforts to integrate languages and models. We discuss these concerns in more detail in Section 4, motivating the need for its further research.

Even when using a common language to express the syntax and semantics of each language, bridges between the different languages and models still need to be built. To enable the creation of such bridges between modelling languages, it has been suggested in [26] to use ontologies. The approach outlined in [25] suggests using ontologies to make the *inherent semantics* [25] behind the language and model constructs explicit, and that way considerably reduce mapping complexity and ambiguity.

3.3 Family of languages

One can take this idea of using a common ontology a step further, by (re)designing the different purpose-specific modelling languages as specializations of a common generic meta-model. This leads to an approach in between the *unified languages* and *federated languages* approaches, which might be called a *family of languages*. Note that such a

generic meta-model is *not* a meta-meta-model. It is a generalized meta-model, where the meta-models of more specific languages can be seen as specialization/refinements of the generic meta-model (i.e. not as instantiations of a meta-meta-model). The idea of using a generic meta-model is akin to older work on the so-called meta-model hierarchy [27], which has also inspired the hierarchy in the meta-model of the ArchiMate language [28].

4 Towards a theory for coherent modelling landscapes

In this section, we develop the first elements of an explanatory theory that aims to gain better insight into the needs and challenges underlying the use of different models/languages in enterprise modelling. We first explore the dimension of domain-specificity in 4.1, by analysing the interplay between domain concepts and normative restrictions on the modelling languages. In 4.2, we discuss the relation between purpose-based model/language tuning, model quality dimensions and identified dimensions of fragmentation.

4.1 Modelling domain and modelling language

To support this analysis, we introduce the matrix as shown in Figure 1. The horizontal dimension of the matrix corresponds to the *openness to different interpretations* of the domain concepts, i.e. (natural language) concepts used to communicate about the domain. Its extremes are defined as:

Open – where multiple interpretations of the domain concepts are possible. This is typically the case in domains with heterogeneous communities, whose practise and use of language differs significantly, resulting in different environments of discourse within the domain [29].

Normative – where there is a single (allowed) interpretation of domain concepts. Typically, domain concepts are defined within the normative documents. For example, in the insurance sector, standardized definitions of the insurance concepts exist within the i.e. Business Glossary[4], and are intended to be used as the basis for communication between industry partners, in the development of services, architectures and applications etc.

The vertical dimension of the matrix represents *the scale of normative restriction of modelling language*. We define the extremes on this scale as follows:

Open – with modelling language not being restricted a priori, but being constructed along the process of domain modelling.

Normative – refers to the modelling languages whose both syntax and semantics are formally defined in a mathematical language, resulting in one possible interpretation of the modelling language constructs.

[4] `http://acord.org/resources/framework/Pages/default.aspx`

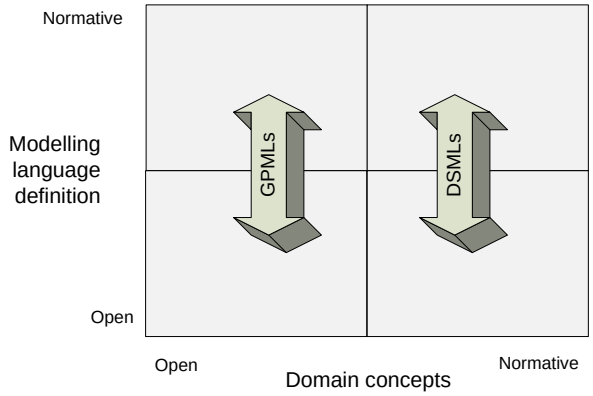

Fig. 1. Language construction view

As already discussed, there is a natural drive towards using domain-specific modelling languages (DSML) in enterprise modelling [12]. Essentially, by incorporating specific concepts tuned to modelling particular problem domains, the ambition of DSMLs is to foster modelling productivity, facilitate the understanding of the models by the stakeholders and increase the overall (in particular semantic and pragmatic) quality of the resulting models. By incorporating domain concepts into modelling language definition, more semantic precision is intended to be given to the language constructs. However, as discussed in [12], the conception of a DSML is bound to complex challenges and contingent decisions, and the level of specificity of the language is a matter of trade-off with the reusability of DSML in different contexts. The more specific DSML is, the more semantic precision it will have, the fewer the number of areas it can be applied to [12]. At the same time, this will increase its conceptual clarity within the intended domain of use, i.e. it will increase the quality model's of socio-cognitive interpretation [9].

DSMLs can be defined and used at various levels of formality, e.g. as semi-formal and visual [5], or executable [30]. The formality of the DSML is a means to improve the quality of technical interpretation [31] of the resulting models. This immediately suggests that the (required) formality of the DSML depends on the intended usage of the resulting models, as will be discussed later.

However, besides DSMLs, rather general-purpose modelling languages (GPML) such as i* [32], e3Value [33] or UML [10] are also used in enterprise modelling practises. They are usually more expressive, but at the same time less suitable than DSMLs in dealing with specific problem domains. The concepts of a GPML are rather generic and their meaning may vary across different contexts of the language use. For example, a modelling language such as i* [32] can be used for modelling strategic goals of actors in relation to the system, but also to express information systems requirements. In each of these contexts, the inherent semantics of e.g. the modelling construct *actor* will vary: when modelling strategic goals of enterprise, actor can only be a human actor, while in the context of modelling software requirements, a machine may be an actor as well.

This context-dependent semantic variation of GPML concepts is to be determined by taking into account the context of its use, i.e. purpose.

Following these discussions, we position DSMLs across the cells on the right side of the matrix, and GPMLs across the cells on the left side of the matrix. The boundary between a GPML and a DSML is not as straightforward, but what clearly distinguishes these two families is the ambition of DSMLs to exclude as much as possible the potential different interpretations of the domain concepts incorporated in the language. While DSMLs offer advantages of zooming in a particular domain with its specific concepts, they also come with the cost, since they emphasize particular terminologies, do not facilitate cross-domain communication, and exert the fragmenting effect on the enterprise modelling landscape. On the other hand, GPMLs are easier to reuse for modelling different domains, however the interpretation challenge of the models expressed in GPMLs is more pronounced.

4.2 Model purpose and model quality

The purpose for which the model is to be used influences the requirements on the information conveyed by the model, but also the way the it should be represented for its intended audience). Therefore, there is a clear connection between the notions of *model purpose* and *model quality* [13, 9]: for a given purpose, different dimensions of model quality are emphasized. This in turn implies different requirements on the modelling language(s) to be used, which ideally is tuned to the purpose at hand.

More specifically, in terms of the SEQUAL framework [13], we see the following relationships between the model quality dimensions and the identified purpose dimensions, which we discussed in Section 2:

Physical quality – links directly to the *restriction of notation* dimension, more specifically referring to the actual medium that is to be used.

Empirical quality – also links to the *restriction of notation* dimension, though referring more to the way the notation is used, e.g. in terms of comprehensibility and readability.

Syntactic quality – links directly to the *restriction of syntax* dimension.

Semantic quality – links directly to the *restriction of semantics* dimension.

Perceived semantic quality – covers the correspondence between actors' interpretation of the model and their current knowledge of the domain. The actor's interpretation of the model will be influenced by a priori choice of the domain and *restrictions on notation, syntax, or semantics*, in particular if they support the actor's prior knowledge and abilities to understand model's representation.

Pragmatic quality – as the correspondence between the model and its interpretation by the audience links to the spectrum of *usage-specific fragmentations*. Combined they enable or restrict the freedom of modellers to *influence* the pragmatic quality of models. In this regard, as argued in [9], we find it useful to distinguish between the quality of socio-cognitive interpretation and the quality of technical interpretation, as language restrictions are differently combined to meet these qualities.

Social quality – is not linked directly to *model purpose* dimension, in our view, but embedded more in the *process* of modelling, and therefore linked to the model*ling* purpose.

Organisational quality – is in our view linked mainly to the *domain-specificity* dimensions, determining the primary goal of modelling in capturing some domain in terms of a model.

We illustrate these considerations with several more or less typical model purposes within enterprise modelling:

Collaborative domain modelling in a heterogeneous community – This variation of domain modelling is typical for the domains with heterogeneous communities, whose practice and use of language differs significantly, resulting in different environments of discourse within the domain [34, 22]. The focus of the modelling process in this context is on reaching conceptual clarity and consensus between participants and the gradual construction of a shared conceptual model. However, given the terminological heterogeneity within the domain, the consensus on the entire vocabulary would not be realistic, so concepts' definitions are not likely to have a normative character. Given the focus on reaching common understanding and agreement, the aspects of the model such as syntactical correctness will be less relevant, therefore, informal and semi-formal modelling notations would be sufficient for this purpose.

Stakeholder communication – In this context, the models are intended to provide various stakeholders with needed insight in issues/problems at hand, and aid in human decision-making. As typically these stakeholders do not have an engineering background, there is a clear preference towards graphical and rather informal or 'box-and-line style' representations [35]. On the other side, the focus on semantic precision of domain concepts used in models is strong, and to avoid misunderstandings, ideally the concepts (and notation) stemming from the stakeholders' professional background would be used. This clearly indicates the tendency towards domain-specificity, i.e. the focus on semantic and pragmatic quality of models.

Model execution – When the model is intended to be directly executable by tools (e.g. specification of the software system), what matters the most is its formality, therefore the focus is on syntactic and semantic quality of the model. The corresponding modelling language therefore should be a formal one, covering at least execution semantics for the model, e.g. [30, 36]. Given that these models are to be used both by human users and machines, the corresponding languages may also combine both textual and visual notations e.g. [30]. However, they need not necessarily be domain-specific.

5 Conclusion

In this paper we explored the issues involved in assuring the coherence of enterprise modelling landscapes. We argued that while there is a clear need to have coherence among the set of models used to represent different aspects of the same (version of) an enterprise, the purpose-specificity which enables the tuning of models (and languages) to the purpose at hand, has a fragmenting effect on the modelling landscape. We concluded that the polarity between coherence and purpose-specificity should therefore be

managed carefully. In this context, we discussed some of the existing strategies to reach integrated use of models and languages in enterprise modelling, including their trade-offs. Finally, we sketched some elements of an explanatory theory to better understand the use of enterprise models/languages at various level of domain- and usage-specificity. As a next step, we aim to further develop the latter theory, where we will initially focus on *model purpose*, but in later versions also involve *modelling purpose* in the equation. With such an explanatory theory in place, we can then endeavour to develop heuristics to balance the needs for purpose-specificity of models and the need for coherence.

Acknowledgement. This work has been partially sponsored by the *Fonds National de la Recherche Luxembourg* (`www.fnr.lu`), via the PEARL programme.

References

1. Dietz, J.: Enterprise Ontology – Theory and Methodology. Springer, Berlin, Germany (2006)
2. Op 't Land, M., Proper, H., Waage, M., Cloo, J., Steghuis, C.: Enterprise Architecture – Creating Value by Informed Governance. Enterprise Engineering Series. Springer (2008)
3. Wood–Harper, A., Antill, L., Avison, D.: Information Systems Definition: The Multiview Approach. Blackwell, Oxford, United Kingdom (1985)
4. Zachman, J.: A framework for information systems architecture. IBM Systems Journal **26**(3) (1987)
5. Frank, U.: Multi-perspective Enterprise Modeling (MEMO) - Conceptual Framework and Modeling Languages. In: Proceedings of HICSS'02. Volume 3., IEEE (2002)
6. Winter, R., Fischer, R.: Essential layers, artifacts, and dependencies of enterprise architecture. Journal Of Enterprise Architecture **3**(2) (2007) 7–18
7. Wagter, R., Proper, H., Witte, D.: A Practice-Based Framework for Enterprise Coherence. In: Proceedings of PRET 2012. Volume 120 of LNBIP., Springer (2012) To appear.
8. Proper, H., Hoppenbrouwers, S., Veldhuijzen van Zanten, G.: Communication of Enterprise Architectures. [11] 67–82
9. Bommel, P.v., Hoppenbrouwers, S., Proper, H., Roelofs, J.: Concepts and strategies for quality of modeling. In: Innovations in Information Systems Modeling. IGI Publishing (2008)
10. OMG: UML 2.0 Superstructure Specification – Final Adopted Specification. Technical Report ptc/03–08–02 (August 2003)
11. Lankhorst, M., ed.: Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Berlin, Germany (2005)
12. Frank, U.: Some guidelines for the conception of domain-specific modelling languages. In Nüttgens, M., Thomas, O., Weber, B., eds.: EMISA. Volume 190 of LNI., GI (2011) 93–106
13. Krogstie, J., Sindre, G., Jorgensen, H.: Process models representing knowledge for action: a revised quality framework. European Journal of Information Systems **15** (2006) 91–102
14. Vernadat, F.: UEML: Towards a unified enterprise modelling language. International Journal of Production Research **40**(17) (2002) 4309–4321.
15. Anaya, V., Berio, G., Harzallah, M., Heymans, P., Matulevicius, R., Opdahl, A.L., Panetto, H., Verdecho, M.: The unified enterprise modelling language - overview and further work. Computers in Industry **61** (2010) 99–111
16. Van't Wout, J., Waage, M., Hartman, H., Stahlecker, M., Hofman, A.: The Integrated Architecture Framework Explained. Springer, Berlin, Germany (2010)

17. Iacob, M.E., Jonkers, H., Lankhorst, M., Proper, H., Quartel, D.: ArchiMate 2.0 Specification. The Open Group (2012)
18. The Open Group: TOGAF Version 9. Van Haren Publishing, The Netherlands (2009)
19. Hoppenbrouwers, S., Proper, H., Weide, T.v.d.: Understanding the Requirements on Modelling Techniques. In: 17th International Conference on Advanced Information Systems Engineering, CAiSE 2005. Volume 3520 of LNCS., Springer (2005) 262–276
20. Moody, D.: The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering Software Engineering **35**(6) (2009) 756–779
21. Hofstede, A.t., Proper, H.: How to Formalize It? Formalization Principles for Information Systems Development Methods. Information and Software Technology **40**(10) (October 1998) 519–540
22. Hoppenbrouwers, S., Bleeker, A., Proper, H.: Facing the Conceptual Complexities in Business Domain Modeling. Computing Letters **1**(2) (2005) 59–68
23. van der Linden, D.J.T., Hoppenbrouwers, S., Lartseva, A., Proper, H.A.: Towards an investigation of the conceptual landscape of enterprise architecture. In: BMMDS/EMMSAD. Volume 81 of LNBIP., Springer (2011) 526–535
24. Opdahl, A., Berio, G., Harzallah, M., Matulevicius, R.: Ontology for enterprise and information systems modelling. Applied Ontology **7(1)** (2012) 49–92
25. Karagiannis, D., Höfferer, P.: Metamodels in action: An overview. In Filipe, J., Shishkov, B., Helfert, M., eds.: ICSOFT (1), INSTICC Press (2006)
26. Brinkkemper, S., Saeki, M., Harmsen, A.: Meta-modelling based assembly techniques for situational method engineering. Information Systems **24**(3) (1999) 209–228
27. Falkenberg, E., Verrijn–Stuart, A., Voss, K., Hesse, W., Lindgreen, P., Nilsson, B., Oei, J., Rolland, C., Stamper, R., eds.: A Framework of Information Systems Concepts. IFIP WG 8.1 Task Group FRISCO, IFIP (1998)
28. Lankhorst, M., Proper, H., Jonkers, H.: The anatomy of the archimate language. International Journal of Information System Modeling and Design (IJISMD) **1**(1) (2010) 1–32
29. Hoppenbrouwers, S.: Freezing Language; Conceptualisation processes in ICT supported organisations. PhD thesis, University of Nijmegen, Nijmegen, The Netherlands (2003)
30. Kleppe, A.: Towards general purpose, high level, software languages. In: ECMDA-FA. Volume 3748 of LNCS., Springer (2005) 220–238
31. Bommel, P.v., Hoppenbrouwers, S., Proper, H., Weide, T.v.d.: QoMo: A Modelling Process Quality Framework based on SEQUAL. In: Proceedings of the 12th Workshop on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD'07), held in conjunctiun with the 19th Conference on Advanced Information Systems (CAiSE'07), Trondheim, Norway, CEUR Workshop Proceedings (2007) 118–127
32. Yu, E., Mylopoulos, J.: Using goals, rules, and methods to support reasoning in business process reengineering. International Journal of Intelligent Systems in Accounting, Finance and Management **5**(1) (1996) 1—13
33. Gordijn, J., Akkermans, H.: Value based requirements engineering: Exploring innovative e-commerce ideas. Requirements Engineering Journal **8**(2) (2003) 114–134
34. Proper, H., Hoppenbrouwers, S.: Concept Evolution in Information System Evolution. In: Proceedings of CAiSE 2004. (2004) 63–72
35. Malavolta, I., Lago, P., Muccini, H., Pellicone, P., Tang, A.: What industry needs from architectural languages: an industrial study. Technical report, University of L'Aquilla (2012)
36. OMG: Semantics of a foundational subset for executable uml models (fuml), v1.0. Technical report, OMG (2011)